ProbReach: Probabilistic Bounded Reachability for Uncertain Hybrid Systems

Fedor Shmarov, Paolo Zuliani

School of Computing Science, Newcastle University, UK

Introduction

- `ProbReach` – tool for probabilistic reachability analysis in uncertain hybrid systems.
    - Uncertain hybrid systems: *random* and *nondeterministic* parameters.

Introduction

- `ProbReach` – tool for probabilistic reachability analysis in uncertain hybrid systems.
    - Uncertain hybrid systems: *random* and *nondeterministic* parameters.
- Reachability = deciding whether a goal state is reachable.
    - Systems with random parameters: computing *probability of reachability*.
    - Nondeterministic parameters introduce a *probability reachability function*.

Introduction

- `ProbReach` – tool for probabilistic reachability analysis in uncertain hybrid systems.
  - Uncertain hybrid systems: *random* and *nondeterministic* parameters.

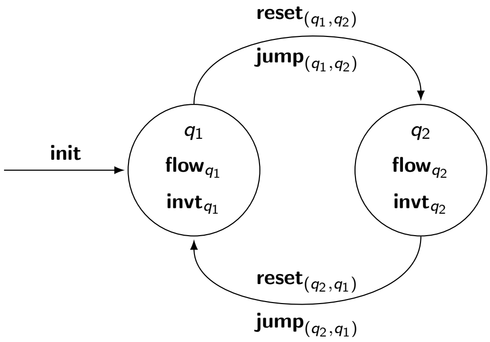
- Reachability = deciding whether a goal state is reachable.
  - Systems with random parameters: computing *probability of reachability*.
  - Nondeterministic parameters introduce a *probability reachability function*.

- `ProbReach` implements two approaches: *formal* and *statistical*.
  - Formal approach: stronger guarantees (*absolute* vs. *statistical*).
  - Statistical approach: lower complexity with respect to the number of parameters (*constant* vs. *exponential*).

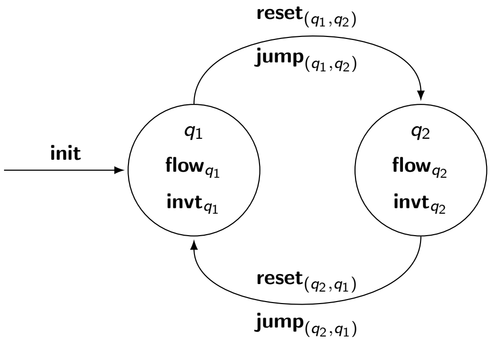
Introduction

- `ProbReach` – tool for probabilistic reachability analysis in uncertain hybrid systems.
  - Uncertain hybrid systems: *random* and *nondeterministic* parameters.
- Reachability = deciding whether a goal state is reachable.
  - Systems with random parameters: computing *probability of reachability*.
  - Nondeterministic parameters introduce a *probability reachability function*.
- `ProbReach` implements two approaches: *formal* and *statistical*.
  - Formal approach: stronger guarantees (*absolute* vs. *statistical*).
  - Statistical approach: lower complexity with respect to the number of parameters (*constant* vs. *exponential*).
- `ProbReach` can be applied to realistic models.
  - Artificial pancreas model.

Hybrid Systems



Hybrid Systems



- **init** and **reset** – computable functions,
- **flow** – Lipschitz-continuous ODEs,
- **invt** and **jump** – Boolean logic formula $\bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{k_i} \left(f_{i,j}(\mathbf{x}) \circ 0 \right) \right)$,
  - $\circ \in \{>, \geq\}$,
  - $f_{i,j}$ – computable function.

Bounded Reachability

- Reachability is **undecidable** even for linear hybrid systems (Alur, Courcoubetis, Henzinger, Ho. 1993).

Bounded Reachability

- Reachability is **undecidable** even for linear hybrid systems (Alur, Courcoubetis, Henzinger, Ho. 1993).
- Bounded reachability:
  - computable **goal** predicate, and
  - finite reachability depth, and
  - bounded time domain in each mode.

Does the hybrid system reach a **goal** state within a finite number of (discrete) steps?

Bounded Reachability

- Reachability is **undecidable** even for linear hybrid systems (Alur, Courcoubetis, Henzinger, Ho. 1993).
- Bounded reachability:
  - computable **goal** predicate, and
  - finite reachability depth, and
  - bounded time domain in each mode.

> Does the hybrid system reach a **goal** state within a finite number of (discrete) steps?

- Nonlinear arithmetics (with trigonometric functions) over the reals is **undecidable** (Tarski, 1951).

Bounded Reachability

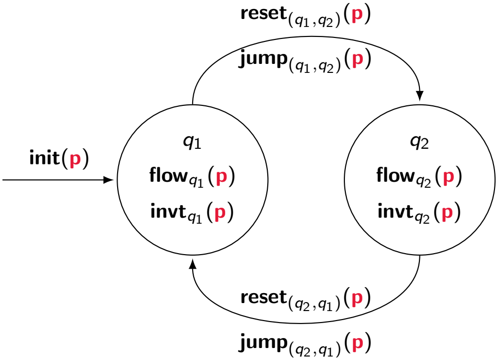
- Reachability is **undecidable** even for linear hybrid systems (Alur, Courcoubetis, Henzinger, Ho. 1993).
- Bounded reachability:
  - computable **goal** predicate, and
  - finite reachability depth, and
  - bounded time domain in each mode.

Does the hybrid system reach a **goal** state within a finite number of (discrete) steps?

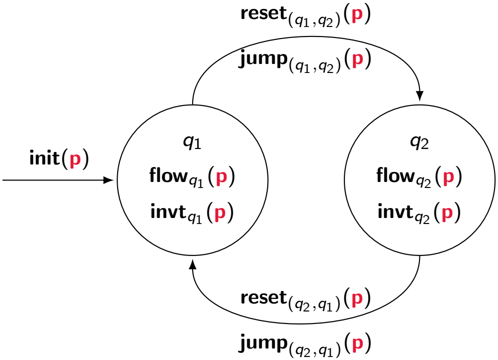
- Nonlinear arithmetics (with trigonometric functions) over the reals is **undecidable** (Tarski, 1951).

- Bounded reachability is δ-**decidable**.
  - δ-complete decision procedure (Gao, Avigad, Clarke. LICS 2012).

Uncertain Hybrid Systems



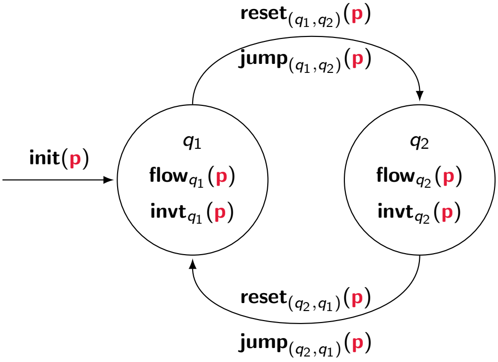
Uncertain Hybrid Systems



Parametric Hybrid System (PHS):

- $\mathbf{p} \in P$ – parameter,
- $P \neq \emptyset$ – parameter space,
- $\frac{d\mathbf{p}}{dt} = 0$.

Uncertain Hybrid Systems



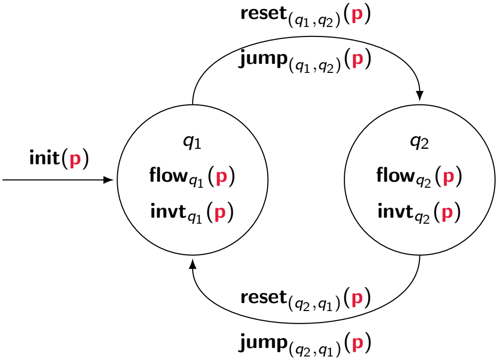
Parametric Hybrid System (PHS):

- $\mathbf{p} \in P$ – parameter,
- $P \neq \emptyset$ – parameter space,
- $\frac{d\mathbf{p}}{dt} = 0$.

Stochastic PHS (SPHS):

- PHS with random parameters.

Uncertain Hybrid Systems



Parametric Hybrid System (PHS):

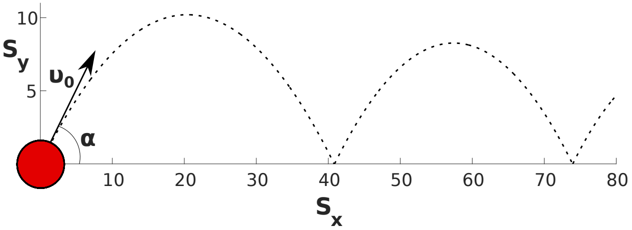
- $\mathbf{p} \in P$ – parameter,
- $P \neq \emptyset$ – parameter space,
- $\frac{d\mathbf{p}}{dt} = 0$.

Stochastic PHS (SPHS):

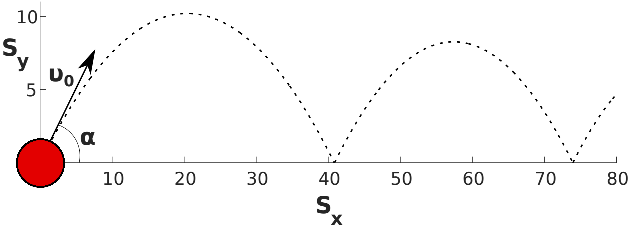
- PHS with random parameters.

- **init** and **reset** – computable functions,
- **flow** – Lipschitz-continuous ODEs,
- **invt** and **jump** – Boolean logic formula $\bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{k_i} \left(f_{i,j}(\mathbf{x}, \mathbf{p}) \circ 0 \right) \right)$,
  - $\circ \in \{>, \geq\}$,
  - $f_{i,j}$ – computable function.

SPHS: Running Example



SPHS: Running Example



Parameters:

- **Random**:
  - $v_0 \sim \mathcal{N}(25, 3)$ – initial speed,
  - $\alpha = \{0.7854 : 0.9, 1.0472 : 0.09, 0.5236 : 0.01\}$ – angle to horizon,
- **Nondeterministic**:
  - $K \in [0.5, 0.9]$ – speed loss coefficient.

Bounded Reachability Probability

What is the probability that the system reaches a **goal** state in a finite number of steps?

Bounded Reachability Probability

What is the probability that the system reaches a **goal** state in a finite number of steps?

Let $P = P_N \times P_R$

- P_R is the domain of random parameters,
- P_N is the domain of nondeterministic parameters

Bounded Reachability Probability

What is the probability that the system reaches a **goal** state in a finite number of steps?

Let $P = P_N \times P_R$

- P_R is the domain of random parameters,
- P_N is the domain of nondeterministic parameters

The bounded reachability probability function is:

$$\mathbf{Pr} : P_N \to [0, 1].$$

If $P_N = \emptyset$ then **Pr** is constant.

Computing Bounded Reachability Probability

Approach	**Formal**	**Statistical**
Principle	Formal Reasoning	Monte Carlo Sampling
Probability	$\int_G d\mathbb{P}$	$\mathbb{E}[X] \approx \frac{1}{N}\sum_{i=1}^{N} X_i$
	$G = \{\mathbf{p} \in P : \mathbf{goal}(\mathbf{p})\}$, $G^C = P \setminus G$.	$X_i = 1$ if $\mathbf{goal}(\mathbf{p})$, $X_i = 0$ otherwise.
Guarantees	Absolute	Statistical
Complexity (number of parameters)	Exponential	Constant

Computing Bounded Reachability Probability

Approach	**Formal**	**Statistical**
Principle	Formal Reasoning	Monte Carlo Sampling
Probability	$\int_G d\mathbb{P}$	$\mathbb{E}[X] \approx \frac{1}{N}\sum_{i=1}^{N} X_i$
	$G = \{\mathbf{p} \in P : \mathbf{goal}(\mathbf{p})\}$, $G^C = P \setminus G$.	$X_i = 1$ if $\mathbf{goal}(\mathbf{p})$, $X_i = 0$ otherwise.
Guarantees	Absolute	Statistical
Complexity (number of parameters)	Exponential	Constant

Both approaches need a procedure which given a non-empty $B \subseteq P$ identifies whether $B \subseteq G$ or $B \subseteq G^C$.

Evaluation Procedure (I)

Given a non-empty $B \subseteq P$ we define two formulae **Reach**(H, I, B), and **Reach**$^{\forall}(H, I, B)$.

Evaluation Procedure (I)

Given a non-empty $B \subseteq P$ we define two formulae
$\mathbf{Reach}(H, l, B)$, and $\mathbf{Reach}^{\forall}(H, l, B)$.

- $\mathbf{Reach}(H, l, B)$ – true if H satisfies **goal** in l steps for some $\mathbf{p} \in B$,
- $\mathbf{Reach}^{\forall}(H, l, B) := \forall^{B}\mathbf{p} : \mathbf{Reach}(H, l, \{\mathbf{p}\})$.
  - $\mathbf{Reach}^{\forall}(H, l, B) \Rightarrow \mathbf{Reach}(H, l, \{\mathbf{p}\})$.

Evaluation Procedure (I)

Given a non-empty $B \subseteq P$ we define two formulae $\textbf{Reach}(H, l, B)$, and $\textbf{Reach}^{\forall}(H, l, B)$.

- $\textbf{Reach}(H, l, B)$ – true if H satisfies **goal** in l steps for some $\mathbf{p} \in B$,
- $\textbf{Reach}^{\forall}(H, l, B) := \forall^{B}\mathbf{p} : \textbf{Reach}(H, l, \{\mathbf{p}\})$.
  - $\textbf{Reach}^{\forall}(H, l, B) \Rightarrow \textbf{Reach}(H, l, \{\mathbf{p}\})$.
- **Reach** and $\textbf{Reach}^{\forall}$ are bounded $\mathcal{L}_{\mathbb{R}}$-sentences
- Can be verified by the δ-complete decision procedure

Evaluation Procedure (I)

Given a non-empty $B \subseteq P$ we define two formulae **Reach**(H, l, B), and **Reach**$^\forall(H, l, B)$.

- **Reach**(H, l, B) – true if H satisfies **goal** in l steps for some $\mathbf{p} \in B$,
- **Reach**$^\forall(H, l, B) := \forall^B \mathbf{p}$: **Reach**$(H, l, \{\mathbf{p}\})$.
  - **Reach**$^\forall(H, l, B) \Rightarrow$ **Reach**$(H, l, \{\mathbf{p}\})$.
- **Reach** and **Reach**$^\forall$ are bounded $\mathcal{L}_{\mathbb{R}}$-sentences
- Can be verified by the δ-complete decision procedure

Remember!!! Only *unsat* answer can be trusted and δ-*sat* is subject to over-approximation δ.

Evaluation Procedure (II)

- Based on the *unsat* (trusted) answer of δ-decision procedures

Algorithm 1: evaluate(H, I, B, δ)

if δ-*decision*$\Big(\textbf{Reach}(H, I, B)\Big) == \delta$-*sat* **then**
  if δ-*decision*$\Big(\neg\textbf{Reach}^{\forall}(H, I, B)\Big) == \delta$-*sat* **then**
    return undet;
  return sat;
return unsat;

- **sat** – **goal** is reached for **all** parameter values in B,
- **unsat** – **goal** is reached for **no** parameter values in B,
- **undet** – **goal** is reached for **some** parameter values in B,

Evaluation Procedure (II)

- Based on the *unsat* (trusted) answer of δ-decision procedures

Algorithm 2: evaluate(H, I, B, δ)

if δ-*decision*$\Big($**Reach**$(H, I, B)\Big) == \delta$-*sat* **then**
    if δ-*decision*$\Big(\neg$**Reach**$^{\forall}(H, I, B)\Big) == \delta$-*sat* **then**
        return undet;
    return sat;
return unsat;

- **sat** – **goal** is reached for **all** parameter values in B,
- **unsat** – **goal** is reached for **no** parameter values in B,
- **undet** – **goal** is reached for **some** parameter values in B,

OR

- **undet** – one of the formulae is not robust for the given δ.

Computing Bounded Reachability Probability

Approach	**Formal**	**Statistical**
Principle	Formal Reasoning	Monte Carlo Sampling
Probability	$\int_G d\mathbb{P}$	$\mathbb{E}[X] = \frac{1}{N}\sum_{i=1}^{N} X_i$
	$G = \{\mathbf{p} \in P : \mathbf{goal}(\mathbf{p})\}$, $G^C = P \setminus G$.	$X_i = 1$ if $\mathbf{goal}(\mathbf{p})$, $X_i = 0$ otherwise.
Guarantees	Absolute	Statistical
Complexity (number of parameters)	Exponential	Constant

Formal Approach: “In a nutshell”

The bounded reachability probability is a function of **nondeterministic** parameters obtained as:

$$\mathbf{Pr}(\mathbf{p}_N) = \int_{G(\mathbf{p}_N)} d\mathbb{P}$$

- $\mathbb{P}$ - probability measure of random parameters,
- $G(\mathbf{p}_N)$ - system's goal set for the given $\mathbf{p}_N$.

Formal Approach: “In a nutshell”

The bounded reachability probability is a function of **nondeterministic** parameters obtained as:

$$\mathbf{Pr}(\mathbf{p}_N) = \int_{G(\mathbf{p}_N)} d\mathbb{P}$$

- $\mathbb{P}$ - probability measure of random parameters,
- $G(\mathbf{p}_N)$ - system's goal set for the given $\mathbf{p}_N$.

How to compute $\mathbf{Pr}(\mathbf{p}_N)$???

Formal Approach: "In a nutshell"

The bounded reachability probability is a function of **nondeterministic** parameters obtained as:

$$\mathbf{Pr}(\mathbf{p}_N) = \int_{G(\mathbf{p}_N)} d\mathbb{P}$$

- $\mathbb{P}$ - probability measure of random parameters,
- $G(\mathbf{p}_N)$ - system's goal set for the given $\mathbf{p}_N$.

How to compute $\mathbf{Pr}(\mathbf{p}_N)$???

- Identify $G(\mathbf{p}_N)$ – already solved!
  - Partition P_R with boxes B,
  - Evaluate each $\{\mathbf{p}_N\} \times B$ using procedure **evaluate**.

Formal Approach: "In a nutshell"

The bounded reachability probability is a function of **nondeterministic** parameters obtained as:

$$\mathbf{Pr}(\mathbf{p}_N) = \int\limits_{G(\mathbf{p}_N)} d\mathbb{P}$$

- $\mathbb{P}$ - probability measure of random parameters,
- $G(\mathbf{p}_N)$ - system's goal set for the given $\mathbf{p}_N$.

How to compute $\mathbf{Pr}(\mathbf{p}_N)$???

- Identify $G(\mathbf{p}_N)$ – already solved!
  - Partition P_R with boxes B,
  - Evaluate each $\{\mathbf{p}_N\} \times B$ using procedure **evaluate**.
- Compute $\int\limits_B d\mathbb{P}$ for each box with desired precision $\hat{\epsilon} > 0$.
  - Find an estimate which is at most $\hat{\epsilon}$ far from $\int\limits_B d\mathbb{P}$.

Formal Approach: Algorithm

We reason about parameter boxes $B_N \subseteq P_N$ for which we compute enclosures $\left[\mathbf{P}_{over}[B_N], \mathbf{P}_{under}[B_N]\right]$ such that:

$$\forall \mathbf{p}_N \in B_N : \mathbf{Pr}(\mathbf{p}_N) \in \left[\mathbf{P}_{over}[B_N], \mathbf{P}_{under}[B_N]\right].$$

Formal Approach: Algorithm

We reason about parameter boxes $B_N \subseteq P_N$ for which we compute enclosures $\left[\mathbf{P}_{over}[B_N], \mathbf{P}_{under}[B_N]\right]$ such that:

$$\forall \mathbf{p}_N \in B_N : \mathbf{Pr}(\mathbf{p}_N) \in \left[\mathbf{P}_{over}[B_N], \mathbf{P}_{under}[B_N]\right].$$

$B_N = P_N; B_R = P_R;$

$\mathbf{P}_{over}[B_N] = 1; \mathbf{P}_{under}[B_N] = 0;$

while for each B_N: $(\mathbf{P}_{over}[B_N] - \mathbf{P}_{under}[B_N] > \epsilon)$ *or* $(B_N > \rho)$ **do**

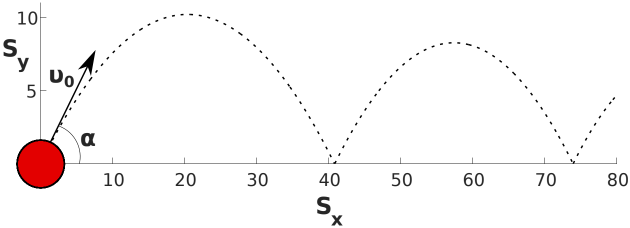
  switch evaluate$(H, I, B_R \times B_N, |B_R|)$ **do**

    case unsat do $\mathbf{P}_{over}[B_N] = \mathbf{P}_{over}[B_N] - \int_{B_R} d\mathbb{P}$;

    case sat do $\mathbf{P}_{under}[B_N] = \mathbf{P}_{under}[B_N] + \int_{B_R} d\mathbb{P}$;

    case undet do bisect B_R, B_N ;

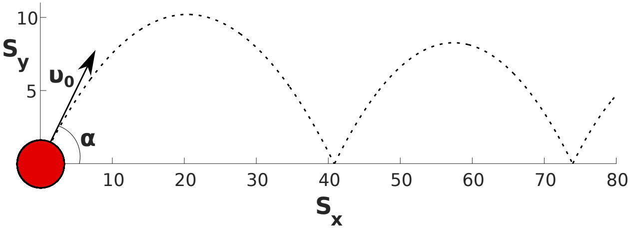
Formal Approach: Running Example



Parameters:

- **Random**:
  - $v_0 \sim \mathcal{N}(25, 3)$ – initial speed,
  - $\alpha = \{0.7854 : 0.9, 1.0472 : 0.09, 0.5236 : 0.01\}$ – angle to horizon,
- **Nondeterministic**:
  - $K \in [0.5, 0.9]$ – speed loss coefficient.

Formal Approach: Running Example



Parameters:

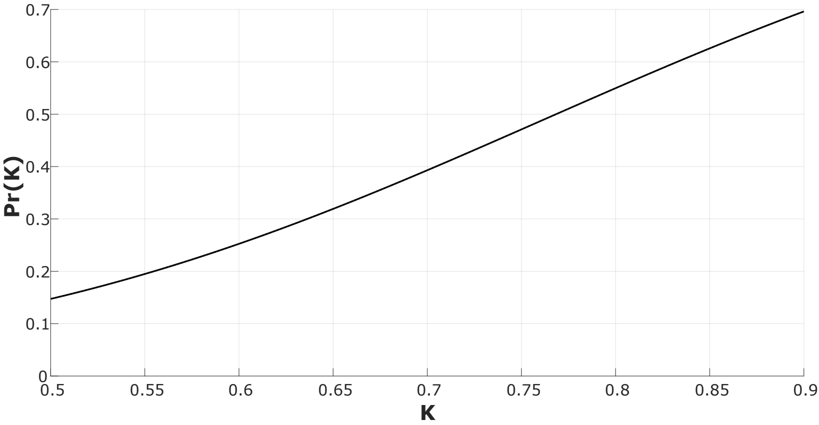
- **Random**:
  - $\upsilon_0 \sim \mathcal{N}(25, 3)$ – initial speed,
  - $\alpha = \{0.7854 : 0.9, 1.0472 : 0.09, 0.5236 : 0.01\}$ – angle to horizon,
- **Nondeterministic**:
  - $K \in [0.5, 0.9]$ – speed loss coefficient.

Compute the probability ($\mathbf{Pr} : [0.5, 0.9] \to [0, 1]$) of landing further than 100 metres ($S_x \geq 100$) after bouncing once ($I = 1$).

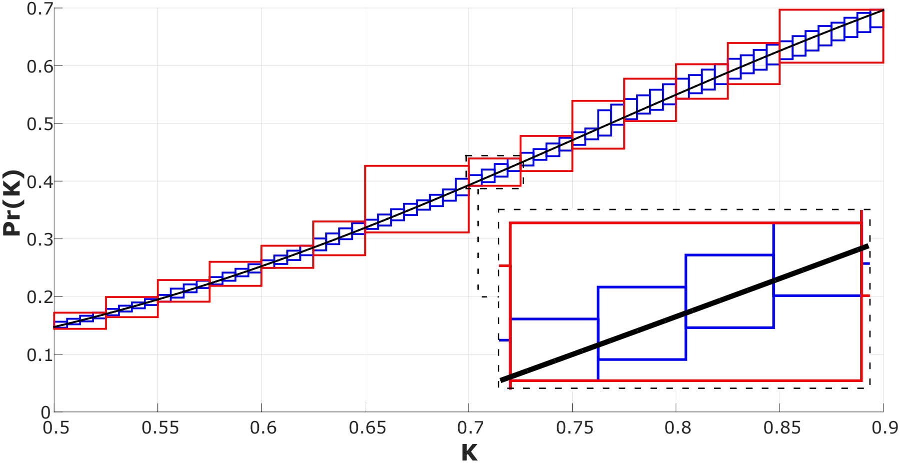
Formal Approach: Running Example (II)

- The probability reachability function $\mathbf{Pr}(K)$ can be obtained as:

$$\mathbf{Pr}(K) = \sum_{i=1}^{3} \left[f_\alpha(\alpha_i) \cdot \int_{\sqrt{\frac{980}{\sin(2\alpha_i)(K^2+1)}}}^{\infty} f_{\upsilon_0}(x)dx \right]$$



Formal Approach: Running Example (III)



- Probability enclosure precision $\epsilon = 10^{-3}$.
- **Red** boxes – computed for $\rho = 5 \cdot 10^{-2}$.
- **Blue** boxes – computed for $\rho = 10^{-2}$.

Formal Approach: ϵ-guarantee

- Size of probability enclosures depends on
  - nondeterministic parameter precision ρ,
  - solver precision δ.

Formal Approach: ϵ-guarantee

- Size of probability enclosures depends on
  - nondeterministic parameter precision ρ,
  - solver precision δ.
- Probability enclosures can be arbitrarily tight (up to the required $\epsilon > 0$) if
  - formulae **Reach** and **Reach**$^{\forall}$ are robust for all $\mathbf{p} \in P$,
  - reachability probability function is continuous,
  - at least one continuous random parameter.

Formal Approach: ϵ-guarantee

- Size of probability enclosures depends on
  - nondeterministic parameter precision ρ,
  - solver precision δ.
- Probability enclosures can be arbitrarily tight (up to the required $\epsilon > 0$) if
  - formulae **Reach** and **Reach**$^{\forall}$ are robust for all $\mathbf{p} \in P$,
  - reachability probability function is continuous,
  - at least one continuous random parameter.

$B_N = P_N; B_R = P_R;$

$\mathbf{P}_{over}[B_N] = 1; \mathbf{P}_{under}[B_N] = 0;$

while for each B_N: $(\mathbf{P}_{over}[B_N] - \mathbf{P}_{under}[B_N] > \epsilon)$ ~~or ($B_N > \rho$)~~ **do**

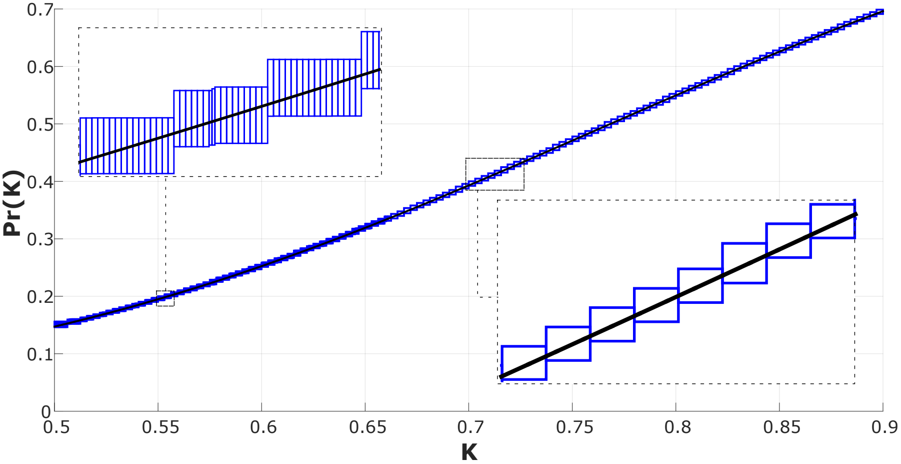
  switch evaluate$(H, I, B_R \times B_N, |B_R|)$ **do**

    case unsat do $\mathbf{P}_{over}[B_N] = \mathbf{P}_{over}[B_N] - \int_{B_R} d\mathbb{P}$;

    case sat do $\mathbf{P}_{under}[B_N] = \mathbf{P}_{under}[B_N] + \int_{B_R} d\mathbb{P}$;

    case undet do bisect B_R, B_N ;

Formal Approach: Running Example (IV)



- Probability enclosure precision $\epsilon = 10^{-2}$.
- Nondeterministic parameter precision ρ is ignored.

Computing Probabilistic Bounded Reachability

Approach	**Formal**	**Statistical**
Principle	Formal Reasoning	Monte Carlo Sampling
Probability	$\int_G d\mathbb{P}$	$\mathbb{E}[X] = \frac{1}{N}\sum_{i=1}^{N} X_i$
	$G = \{\mathbf{p} \in P : \mathbf{goal}(\mathbf{p})\}$, $G^C = P \setminus G$.	$X_i = 1$ if $\mathbf{goal}(\mathbf{p})$, $X_i = 0$ otherwise.
Guarantees	Absolute	Statistical
Complexity (number of parameters)	Exponential	Constant

Statistical Approach: “In a nutshell”

For each $\mathbf{p}_N \in P_N$ and $\mathbf{p}_R \in P_R$ let:

$$X(\mathbf{p}_N, \mathbf{p}_R) = \begin{cases} 1 & \text{if } \textbf{goal} \text{ is reached for } (\mathbf{p}_N, \mathbf{p}_R), \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{Then } \mathbf{Pr}(\mathbf{p}_N) = \mathbb{E}[X(\mathbf{p}_N)] = \int_{G(\mathbf{p}_N)} d\mathbb{P}.$$

Statistical Approach: "In a nutshell"

For each $\mathbf{p}_N \in P_N$ and $\mathbf{p}_R \in P_R$ let:

$$X(\mathbf{p}_N, \mathbf{p}_R) = \begin{cases} 1 & \text{if } \textbf{goal} \text{ is reached for } (\mathbf{p}_N, \mathbf{p}_R), \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{Then } \mathbf{Pr}(\mathbf{p}_N) = \mathbb{E}[X(\mathbf{p}_N)] = \int_{G(\mathbf{p}_N)} d\mathbb{P}.$$

How to compute $\mathbf{Pr}(\mathbf{p}_N)$???

Statistical Approach: "In a nutshell"

For each $\mathbf{p}_N \in P_N$ and $\mathbf{p}_R \in P_R$ let:

$$X(\mathbf{p}_N, \mathbf{p}_R) = \begin{cases} 1 & \text{if } \textbf{goal} \text{ is reached for } (\mathbf{p}_N, \mathbf{p}_R), \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{Then } \mathbf{Pr}(\mathbf{p}_N) = \mathbb{E}[X(\mathbf{p}_N)] = \int_{G(\mathbf{p}_N)} d\mathbb{P}.$$

How to compute $\mathbf{Pr}(\mathbf{p}_N)$???

- Sample $\mathbf{p}_R$ using the parameters' distribution.
- Evaluate $X(\mathbf{p}_N, \mathbf{p}_R)$.

Statistical Approach: “In a nutshell”

For each $\mathbf{p}_N \in P_N$ and $\mathbf{p}_R \in P_R$ let:

$$X(\mathbf{p}_N, \mathbf{p}_R) = \begin{cases} 1 & \text{if } \textbf{goal} \text{ is reached for } (\mathbf{p}_N, \mathbf{p}_R), \\ 0 & \text{otherwise.} \end{cases}$$

Then $\mathbf{Pr}(\mathbf{p}_N) = \mathbb{E}[X(\mathbf{p}_N)] = \int_{G(\mathbf{p}_N)} d\mathbb{P}$.

How to compute $\mathbf{Pr}(\mathbf{p}_N)$???

- Sample $\mathbf{p}_R$ using the parameters’ distribution.
- Evaluate $X(\mathbf{p}_N, \mathbf{p}_R)$.

We CANNOT evaluate $X(\mathbf{p}_N, \mathbf{p}_R)$ (*undecidability !!!*)

Statistical Approach: Confidence Intervals

- We define two random variables:

$$X_{sat}(\mathbf{p}_N, \mathbf{p}_R) = \begin{cases} 1 & \text{if } \textbf{evaluate}(H, I, \{\mathbf{p}_N, \mathbf{p}_R\}, \delta) = \textbf{sat}, \\ 0 & \text{otherwise.} \end{cases}$$

$$X_{usat}(\mathbf{p}_N, \mathbf{p}_R) = \begin{cases} 0 & \text{if } \textbf{evaluate}(H, I, \{\mathbf{p}_N, \mathbf{p}_R\}, \delta) = \textbf{unsat}, \\ 1 & \text{otherwise.} \end{cases}$$

Statistical Approach: Confidence Intervals

- We define two random variables:

$$X_{sat}(\mathbf{p}_N, \mathbf{p}_R) = \begin{cases} 1 & \text{if } \textbf{evaluate}(H, I, \{\mathbf{p}_N, \mathbf{p}_R\}, \delta) = \textbf{sat}, \\ 0 & \text{otherwise.} \end{cases}$$

$$X_{usat}(\mathbf{p}_N, \mathbf{p}_R) = \begin{cases} 0 & \text{if } \textbf{evaluate}(H, I, \{\mathbf{p}_N, \mathbf{p}_R\}, \delta) = \textbf{unsat}, \\ 1 & \text{otherwise.} \end{cases}$$

- $X_{sat}(\mathbf{p}_N, \mathbf{p}_R)$ and $X_{usat}(\mathbf{p}_N, \mathbf{p}_R)$ can be sampled,
- $X_{sat}(\mathbf{p}_N, \mathbf{p}_R) \leq X(\mathbf{p}_N, \mathbf{p}_R) \leq X_{usat}(\mathbf{p}_N, \mathbf{p}_R)$.

$$\mathbb{E}[X_{sat}(\mathbf{p}_N)] \ \leq \ \mathbb{E}[X(\mathbf{p}_N)] = \mathbf{Pr}(\mathbf{p}_N) \ \leq \ \mathbb{E}[X_{usat}(\mathbf{p}_N)]$$

Statistical Approach: Confidence Intervals (II)

- Given accuracy $\xi > 0$ and confidence $c \in (0, 1)$ compute intervals $[p_{sat} - \xi, p_{sat} + \xi]$ and $[p_{usat} - \xi, p_{usat} + \xi]$.
  - $\mathit{Probability}\Big(\mathbb{E}[X_{sat}(\mathbf{p}_N, \mathbf{p}_R)] \in [p_{sat} - \xi, p_{sat} + \xi]\Big) \geq c$,
  - $\mathit{Probability}\Big(\mathbb{E}[X_{usat}(\mathbf{p}_N, \mathbf{p}_R)] \in [p_{usat} - \xi, p_{usat} + \xi]\Big) \geq c$.

Statistical Approach: Confidence Intervals (II)

- Given accuracy $\xi > 0$ and confidence $c \in (0, 1)$ compute intervals $[p_{sat} - \xi, p_{sat} + \xi]$ and $[p_{usat} - \xi, p_{usat} + \xi]$.
  - $Probability\Big(\mathbb{E}[X_{sat}(\mathbf{p}_N, \mathbf{p}_R)] \in [p_{sat} - \xi, p_{sat} + \xi]\Big) \geq c$,
  - $Probability\Big(\mathbb{E}[X_{usat}(\mathbf{p}_N, \mathbf{p}_R)] \in [p_{usat} - \xi, p_{usat} + \xi]\Big) \geq c$.

$$Probability\Big(\mathbf{Pr}(\mathbf{p}_N) \in [p_{sat} - \xi, p_{usat} + \xi]\Big) \geq c.$$

- The size of $[p_{sat} - \xi, p_{usat} + \xi]$ can be greater than 2ξ
  - non-robustness for the given δ, or
  - undecidability in general.

Shmarov and Zuliani. HVC 2016

Statistical Approach: Cross-Entropy Algorithm

- We compute maximum/minimum reachability probability.
  - **approximate** value $\mathbf{p}_N$ where the minimum/maximum probability is achieved,
  - $Probability\big(\mathbf{Pr}(\mathbf{p}_N) \in [p_{sat} - \xi, p_{usat} + \xi]\big) \geq c$.

Statistical Approach: Cross-Entropy Algorithm

- We compute maximum/minimum reachability probability.
  - **approximate** value $\mathbf{p}_N$ where the minimum/maximum probability is achieved,
  - $\mathit{Probability}\big(\mathbf{Pr}(\mathbf{p}_N) \in [p_{sat} - \xi, p_{usat} + \xi]\big) \geq c$.
- CE aims at obtaining the optimal parameter distribution [Rubinstein and Kroese, 2008].

Statistical Approach: Cross-Entropy Algorithm

- We compute maximum/minimum reachability probability.
  - **approximate** value $\mathbf{p}_N$ where the minimum/maximum probability is achieved,
  - $Probability\big(\mathbf{Pr}(\mathbf{p}_N) \in [p_{sat} - \xi, p_{usat} + \xi]\big) \geq c$.

- CE aims at obtaining the optimal parameter distribution [Rubinstein and Kroese, 2008].
  - Step 0 start with **nondeterministic** parameters distributed with PDF $f(\cdot; \mathbf{v})$.

Statistical Approach: Cross-Entropy Algorithm

- We compute maximum/minimum reachability probability.
  - **approximate** value $\mathbf{p}_N$ where the minimum/maximum probability is achieved,
  - $\textit{Probability}\big(\mathbf{Pr}(\mathbf{p}_N) \in [p_{sat} - \xi, p_{usat} + \xi]\big) \geq c$.

- CE aims at obtaining the optimal parameter distribution [Rubinstein and Kroese, 2008].
  - Step 0 start with **nondeterministic** parameters distributed with PDF $f(\cdot; \mathbf{v})$.
  - Step 1 generate samples $\mathbf{p}_N$ using $f(\cdot; \mathbf{v})$.

Statistical Approach: Cross-Entropy Algorithm

- We compute maximum/minimum reachability probability.
  - **approximate** value $\mathbf{p}_N$ where the minimum/maximum probability is achieved,
  - $\mathit{Probability}\big(\mathbf{Pr}(\mathbf{p}_N) \in [p_{sat} - \xi, p_{usat} + \xi]\big) \geq c$.

- CE aims at obtaining the optimal parameter distribution [Rubinstein and Kroese, 2008].
  - Step 0 start with **nondeterministic** parameters distributed with PDF $f(\cdot; \mathbf{v})$.
  - Step 1 generate samples $\mathbf{p}_N$ using $f(\cdot; \mathbf{v})$.
  - Step 2 update $\mathbf{v}$ using the *fittest* (probability-wise) samples.

Statistical Approach: Cross-Entropy Algorithm

- We compute maximum/minimum reachability probability.
  - **approximate** value $\mathbf{p}_N$ where the minimum/maximum probability is achieved,
  - $\mathit{Probability}\big(\mathbf{Pr}(\mathbf{p}_N) \in [p_{sat} - \xi, p_{usat} + \xi]\big) \geq c$.

- CE aims at obtaining the optimal parameter distribution [Rubinstein and Kroese, 2008].
  - Step 0 start with **nondeterministic** parameters distributed with PDF $f(\cdot;\mathbf{v})$.
  - Step 1 generate samples $\mathbf{p}_N$ using $f(\cdot;\mathbf{v})$.
  - Step 2 update $\mathbf{v}$ using the *fittest* (probability-wise) samples.
  - Step 3 go to Step 1 if $f(\cdot;\mathbf{v})$ variance is greater than the desired $\hat{\sigma}^2$.

Statistical Approach: Cross-Entropy Algorithm

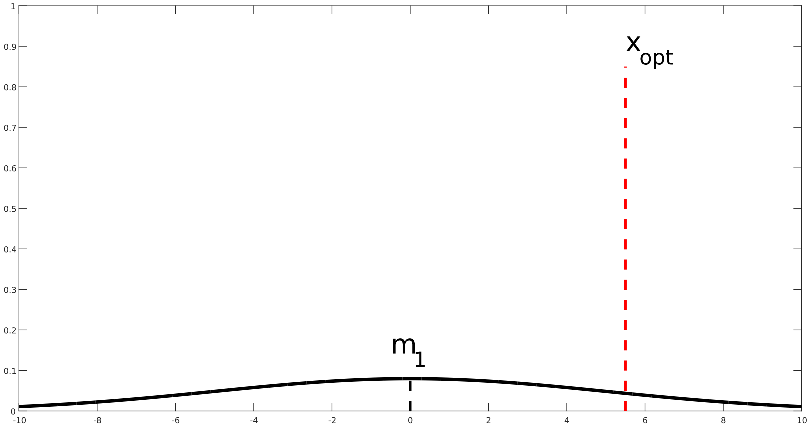
- We compute maximum/minimum reachability probability.
  - **approximate** value $\mathbf{p}_N$ where the minimum/maximum probability is achieved,
  - $\mathit{Probability}\big(\mathbf{Pr}(\mathbf{p}_N) \in [p_{sat} - \xi, p_{usat} + \xi]\big) \geq c$.
- CE aims at obtaining the optimal parameter distribution [Rubinstein and Kroese, 2008].
  - Step 0 start with **nondeterministic** parameters distributed with PDF $f(\cdot; \mathbf{v})$.
  - Step 1 generate samples $\mathbf{p}_N$ using $f(\cdot; \mathbf{v})$.
  - Step 2 update $\mathbf{v}$ using the *fittest* (probability-wise) samples.
  - Step 3 go to Step 1 if $f(\cdot; \mathbf{v})$ variance is greater than the desired $\hat{\sigma}^2$.
- There must be $\mathbf{v}$ such that $f(\cdot; \mathbf{v})$ approximates single-point distributions arbitrarily well.

Statistical Approach: Cross-Entropy Algorithm

- We compute maximum/minimum reachability probability.
  - **approximate** value $\mathbf{p}_N$ where the minimum/maximum probability is achieved,
  - $\mathit{Probability}\big(\mathbf{Pr}(\mathbf{p}_N) \in [p_{sat} - \xi, p_{usat} + \xi]\big) \geq c$.
- CE aims at obtaining the optimal parameter distribution [Rubinstein and Kroese, 2008].
  - Step 0 start with **nondeterministic** parameters distributed with PDF $f(\cdot;\mathbf{v})$.
  - Step 1 generate samples $\mathbf{p}_N$ using $f(\cdot;\mathbf{v})$.
  - Step 2 update $\mathbf{v}$ using the *fittest* (probability-wise) samples.
  - Step 3 go to Step 1 if $f(\cdot;\mathbf{v})$ variance is greater than the desired $\hat{\sigma}^2$.
- There must be $\mathbf{v}$ such that $f(\cdot;\mathbf{v})$ approximates single-point distributions arbitrarily well.

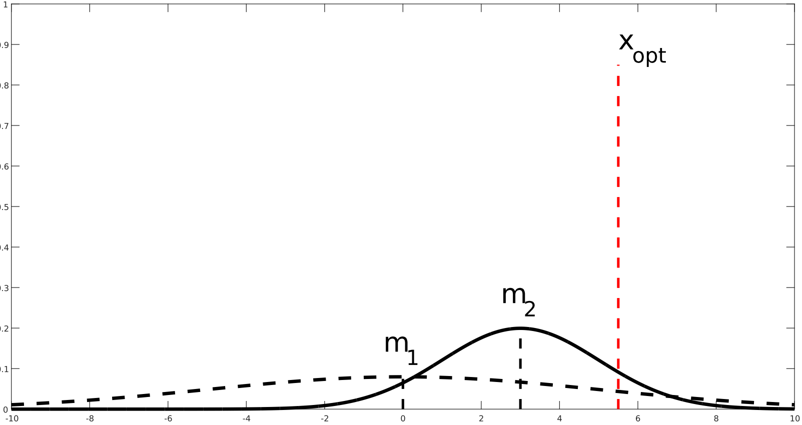
Cross-Entropy can fall into a local extremum.

Statistical Approach: Cross-Entropy Algorithm (II)



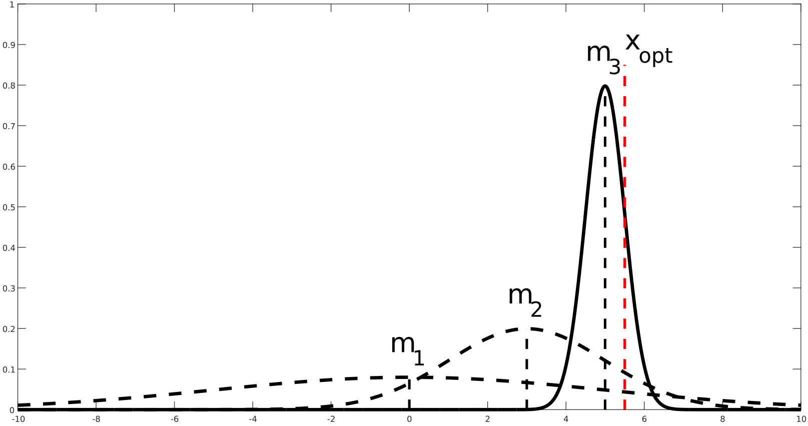
- CE terminates when the user-defined variance $\hat{\sigma}^2$ is reached.

Statistical Approach: Cross-Entropy Algorithm (II)



- CE terminates when the user-defined variance $\hat{\sigma}^2$ is reached.

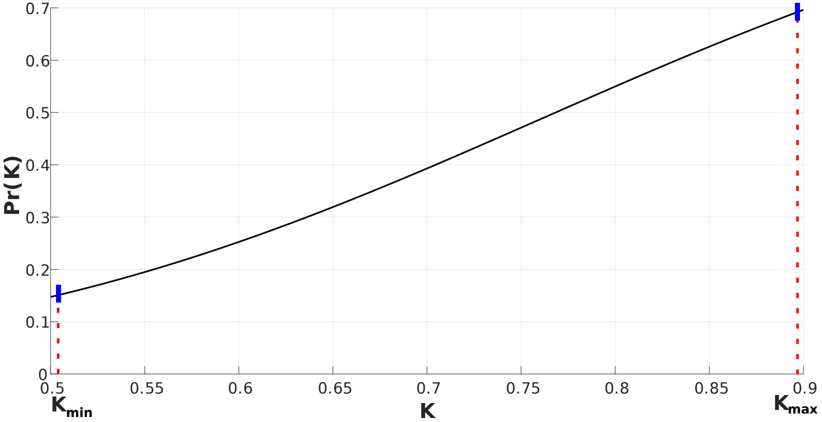
Statistical Approach: Cross-Entropy Algorithm (II)



- CE terminates when the user-defined variance $\hat{\sigma}^2$ is reached.

Statistical Approach: Running Example

- **Pr**(K) can be obtained analytically.



	K	Confidence Interval	**Pr**(K)
min	0.50425	$[0.14464, 0.15464]$	0.15093
max	0.89301	$[0.68238, 0.69238]$	0.68677

Statistical Approach: CE Result Quality

- Number of samples per iteration of CE algorithm,
  - the more the better.

Statistical Approach: CE Result Quality

- Number of samples per iteration of CE algorithm,
  - the more the better.
- Terminal variance value,
  - the smaller the better.

Statistical Approach: CE Result Quality

- Number of samples per iteration of CE algorithm,
  - the more the better.
- Terminal variance value,
  - the smaller the better.
- Initial distribution parameters,
  - need to provide sufficient initial coverage to avoid local extrema.

Statistical Approach: CE Result Quality

- Number of samples per iteration of CE algorithm,
  - the more the better.
- Terminal variance value,
  - the smaller the better.
- Initial distribution parameters,
  - need to provide sufficient initial coverage to avoid local extrema.
- Accuracy for estimating the confidence intervals,
  - the higher the better.

ProbReach

- Implemented in C++.
- Uses OpenMP for parallelisation.
- Uses several libraries
  - CAPD, IBEX, GSL.

[1] http://dreal.github.io/
[2] https://projects.avacs.org/projects/isat3

ProbReach

- Implemented in C++.
- Uses OpenMP for parallelisation.
- Uses several libraries
    - CAPD, IBEX, GSL.
- Any SAT ODE solver supporting δ-decisions can be used.
    - dReal[1] [Sicun Gao, Soonho Kong]
    - iSAT3[2] [Martin Fränzle *et al.*]
- Available at `https://github.com/dreal/probreach`

Shmarov and Zuliani. HSCC 2015

[1] `http://dreal.github.io/`

[2] `https://projects.avacs.org/projects/isat3`

Demonstration

Discussion

- We presented `ProbReach` – tool for probabilistic bounded reachability in uncertain hybrid system.
- It features formal and statistical approaches.
- Formal approach: computes probability enclosures containing the range of the probability reachability function.
  - Complexity grows exponentially with the number of system parameters.
- Statistical approach: computes confidence intervals containing the approximate maximum/minimum probability value.
  - Complexity remains constant with respect to the number of system parameters.
- `ProbReach` is publicly available at `https://github.com/dreal/probreach`.

Automated Synthesis of Safe and Robust PID Controllers for Stochastic Hybrid Systems

Fedor Shmarov[1], Nicola Paoletti[2], Ezio Bartocci[3], Shan Lin[2], Scott A. Smolka[2], Paolo Zuliani[1]

[1]Newcastle University, UK,
[2]Stony Brook University, NY, USA,
[3]TU Wien, Austria

Artificial Pancreas

Closed-loop (with feedback) control of insulin treatment for Type 1 diabetes.

- Continuous glucose monitor
- Control algorithm
- Insulin pump
  - basal – constant dose (automatic)
  - bolus – single high dose (manual)

MINIMED 670G by Medtronic[3]

[3] https://www.medtronicdiabetes.com/products/minimed-670g-insulin-pump-system

Artificial Pancreas

Closed-loop (with feedback) control of insulin treatment for Type 1 diabetes.

- Continuous glucose monitor
- Control algorithm
- Insulin pump
  - basal – constant dose (automatic)
  - bolus – single high dose (manual)

MINIMED 670G by Medtronic[3]

Objective

Design automatic closed-loop control of bolus insulin for keeping blood glucose level between 4-12 mmol/L.

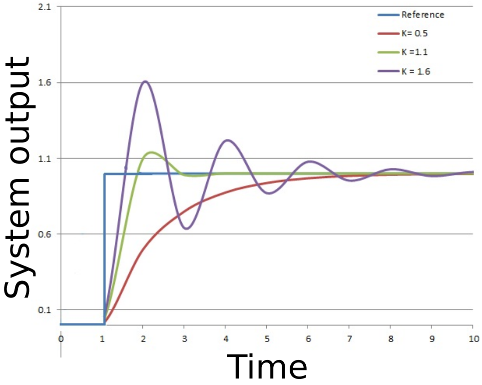
- Temporary **hyperglycemia** is allowed while **hypoglycemia** should be avoided.

[3] https://www.medtronicdiabetes.com/products/minimed-670g-insulin-pump-system

Automatic Control

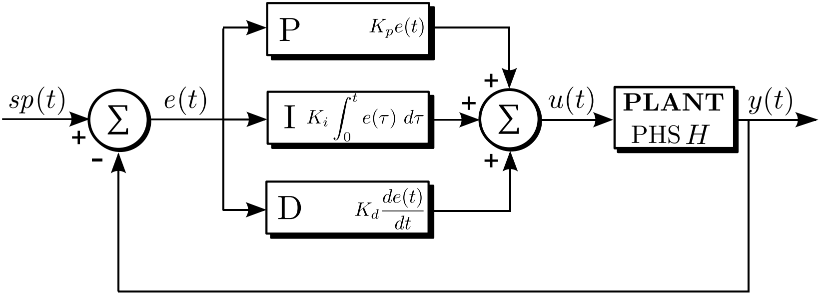
Control Objective

Given an external **disturbance** reduce the **difference** between the measured **system output** and the **desired value** by adjusting the **control variable**.



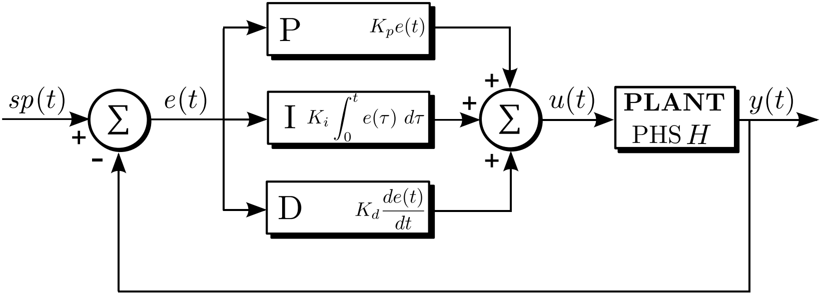
- **disturbance**:
  amount of carbohydrates (D_G)
- **system output**:
  blood glucose level ($G(t)$)
- **desired level** (set-point):
  $G_{sp} = 6.11$ [mmol/L]
- **control variable**:
  insulin admission ($u(t) + u_b$)
- **difference** (error):
  $e(t) = G_{sp} - G(t)$

PID Controller



- P roportional - present value of the error,
- I ntegral - past errors,
- D erivative - predicted future errors.

PID Controller



- $\boxed{P}$roportional - present value of the error,
- $\boxed{I}$ntegral - past errors,
- $\boxed{D}$erivative - predicted future errors.

Synthesis Objective

Find values of K_p, K_i and K_d (gains) "minimising" $e(t)$.

Stochastic Parametric Hybrid Systems

Meal

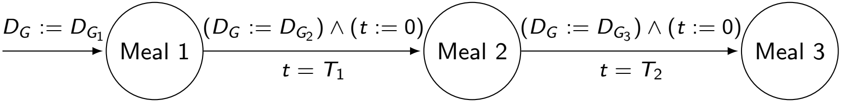
$$\text{Flow}\Big(\underbrace{G(t)}_{\textbf{glucose}}, \underbrace{u(t)+u_b}_{\textbf{insulin}}, \underbrace{G_{sp}}_{\textbf{desired value}}, \underbrace{D_G}_{\textbf{meal size}}\Big)^a.$$

$$\Uparrow \qquad \Downarrow$$

$$\text{PID}(K_p, K_i, K_d, G(t), u(t)+u_b, G_{sp})$$

[a] Hovorka, R.: Closed-loop insulin delivery: from bench to clinical practice. Nature Reviews Endocrinology 7(7), 385395 (2011)

Stochastic Parametric Hybrid Systems (II)



Parameters:

Size of each meal:

$$D_{G_1} \sim \mathcal{N}(40, 10),$$
$$D_{G_2} \sim \mathcal{N}(90, 10),$$
$$D_{G_3} \sim \mathcal{N}(60, 10).$$

Time between the meals:

$$T_1 \sim \mathcal{N}(300, 10),$$
$$T_2 \sim \mathcal{N}(300, 10).$$

Safety and Robustness

Safety

An unsafe state should be reached with very small probability.

Unsafe: $G(t) \notin [4, 16]$.

Safety and Robustness

Safety

An unsafe state should be reached with very small probability.

Unsafe: $G(t) \notin [4, 16]$.

Robustness (not in the sense of δ-robustness)

Difference between the system output and the desired value should be small.

- **Fundamental Index**: $FI(t) = \int_0^t (e(\tau))^2 \, d\tau$

System output should converge to the steady-state.

- **Weighted Fundamental Index**: $FI_w(t) = \int_0^t \tau^2 \cdot (e(\tau))^2 \, d\tau$

Non-robust: $(FI(t) > 3.5 \cdot 10^6) \vee (FI_w(t) > 70 \cdot 10^9)$.

Automated Synthesis

Safety and robustness analysis is performed through **bounded reachability**.

Bounded Reachability

Can the unsafe state be reached within:

- finite number of discrete steps, and
- bounded time interval.

Bounds:

- 3 meals,
- 24 hours.

Automated Synthesis

Safety and robustness analysis is performed through **bounded reachability**.

Bounded Reachability

Can the unsafe state be reached within:

- finite number of discrete steps, and
- bounded time interval.

Bounds:

- 3 meals,
- 24 hours.

Automated Synthesis Objective

Synthesise a PID controller minimizing the probability of reaching an unsafe state or violating the robustness constraint during 3 meals within 24 hour period.

Results

Insulin Administration

$$\underbrace{u(t)}_{\text{PID}(K_p, K_i, K_d, e(t))} + \underbrace{u_b}_{\text{basal rate}}$$

[§] Both safety and robustness were taken into account

Results

Insulin Administration

$$\underbrace{u(t)}_{\text{PID}(K_p, K_i, K_d, e(t))} + \underbrace{u_b}_{\text{basal rate}}$$

Basal rate synthesis (formal): with $G(0) = G_{sp}$ and no external disturbances $G(t)$ reaches $[G_{sp} - 0.05, G_{sp} + 0.05]$ in 2000 minutes and remains there for another 1000 minutes.

	Domain	Result	Chosen Value
u_b	[0,1]	[0.0553359375, 0.055640625]	0.0555

§Both safety and robustness were taken into account

Results

Insulin Administration

$$\underbrace{u(t)}_{\text{PID}(K_p, K_i, K_d, e(t))} + \underbrace{u_b}_{\text{basal rate}}$$

Basal rate synthesis (formal): with $G(0) = G_{sp}$ and no external disturbances $G(t)$ reaches $[G_{sp} - 0.05, G_{sp} + 0.05]$ in 2000 minutes and remains there for another 1000 minutes.

	Domain	Result	Chosen Value
u_b	[0,1]	[0.0553359375, 0.055640625]	0.0555

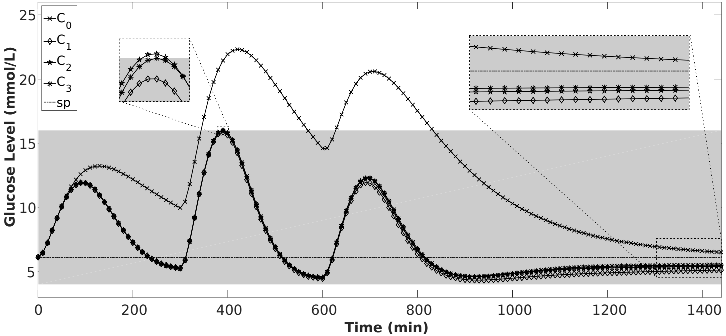
PID controller synthesis (statistical):

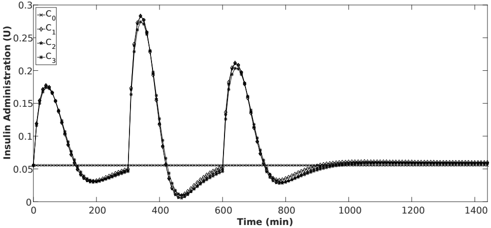
#	K_d	K_i	K_p	CI
C_0^1	0	0	0	$[0.86956, 0.88956]$
C_0^2	0	0	0	$[0.98861, 1]$
C_1	-6.06855×10^{-2}	-5.61901×10^{-7}	-5.979×10^{-4}	$[0.09946, 0.10946]$
C_2	-6.02376×10^{-2}	-3.53308×10^{-7}	-6.166×10^{-4}	$[0.20711, 0.21711]$
C_3[§]	-5.7284×10^{-2}	-3.00283×10^{-7}	-6.39023×10^{-4}	$[0.3324, 0.3524]$

[§] Both safety and robustness were taken into account

One-day Scenario

50 grams, 100 grams, 70 grams in 5 hour intervals.





#	Safety	$FI \times 10^{-6}$	$FI_w \times 10^{-9}$
C_0^1, C_0^2	Unsafe	26.2335	847.5063
C_1	Safe	3.89437	114.49821
C_2	Unsafe	3.95773	81.61823
C_3	Safe	3.96117	74.90655

Discussion

Conclusions:

- We presented a technique for the automated synthesis of safe and robust PID controllers using `ProbReach`.
- The presented approach was applied to an artificial pancreas model.

Future work:

- PID controllers with nonlinear gains.
- Discrete-time PID controllers.

Questions?